



## PATENT ABSTRACTS OF JAPAN

(11) Publication number: **2003189108 A**(43) Date of publication of application: **04.07.03**

(51) Int. Cl.

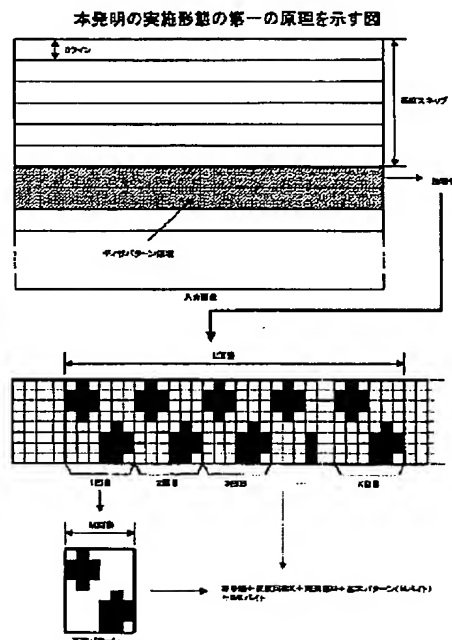
**H04N 1/41**  
**H03M 7/46**
(21) Application number: **2001388931**(22) Date of filing: **21.12.01**(71) Applicant: **FUJITSU LTD**
(72) Inventor: **SAKAI KENICHIRO**  
**NODA TSUGUO**
(54) **IMAGE COMPRESSION METHOD**

## (57) Abstract:

**PROBLEM TO BE SOLVED:** To provide a coding method for improving compression rate by applying coding to a pattern consecutively and repetitively appeared.

**SOLUTION:** The image compression method comprises: scanning a binary image constituting an image for each of 8 lines (rows) and encoding parts not including black pixels while skipping them; detecting a unit pattern of a repetitive pattern among the 8 lines being the scanning unit and counting the number of repetitive times when the lines including black pixels have a dither pattern; then encoding the dither pattern by using the number of repetitive times of the unit pattern, a width (the number of columns) of the unit pattern, and pixel data of the unit pattern.

COPYRIGHT: (C)2003,JPO



(11)特許出願公開番号  
特開2003-189108  
(P2003-189108A)

(51)Int.Cl. <sup>7</sup>	識別記号	F I	テ-マ-ト* (参考)
H 0 4 N 1/41		H 0 4 N 1/41	B 5 C 0 7 8
H 0 3 M 7/46		H 0 3 M 7/46	5 J 0 6 4

[illegible]

## 【特許請求の範囲】

【請求項 1】 画像データを圧縮する方法であって、  
2 値画像データの内、反復パターンが含まれている行を  
抽出する抽出ステップと、  
該反復パターンの単位パターンと、該単位パターンの繰  
り返し回数を検出する反復パターン検出ステップと、  
少なくとも、該単位パターンのデータと、反復パターン  
内における該単位パターンの繰り返し回数とを使って、  
画像データ内の反復パターンを含む行を符号化する符号  
化ステップと、を備えることを特徴とする画像圧縮方  
法。

【請求項 2】 前記単位パターンの画素幅が予め定められ  
た 1 以上の固定値であることを特徴とする請求項 1 に記  
載の画像圧縮方法。

【請求項 3】 前記符号化ステップにおいては、単位パタ  
ーンを更にデータ圧縮し、該圧縮された単位パターンと  
該単位パターンの繰り返し回数により符号化することを  
特徴とする請求項 1 に記載の画像圧縮方法。

【請求項 4】 前記符号化ステップは、ランレングス符号  
化方法を使用することを特徴とする請求項 1 に記載の画  
像圧縮方法。

【請求項 5】 前記符号化ステップにおいて、出現した単  
位パターンを登録するテーブルを生成し、該単位パター  
ンと同じ単位パターンが符号化の過程で 2 回以上出てく  
る場合には、該単位パターンの代わりに該テーブルにお  
ける該単位パターンの登録番号を用いて、画像データを  
符号化することを特徴とする請求項 1 に記載の画像圧縮  
方法。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】 本発明は、2 値のビットマッ  
プ画像の圧縮率を向上させる圧縮方法に関する。

## 【0002】

【従来の技術】 近年、パソコンなどの情報機器が普及  
し、様々な用途に使用されている。特に、パソコンで画  
像を処理する場合が多く見られ、パソコンにプリンタを  
接続し、画像を印刷したり、インターネットなどのネッ  
トワークを介して画像データを転送するなどの処理が行  
われている。しかし、画像が複雑になればなるほど、パ  
ソコンからプリンタへの画像データの転送や、ネットワ  
ークを介しての画像データの転送に時間がかかり過ぎる  
ようになり、画像データを圧縮して転送することが望ま  
れる。画像データを圧縮することによって、送信すべき  
データの量が少なくなり、転送時間を節約できるもので  
ある。

【0003】 この場合、画像データには、カラーのもの  
や、グレースケールのものなどがあるが、以下では、こ  
れらの画像データを 2 値化した 2 値化ビットマップ画像  
の圧縮方法について述べる。

【0004】 従来の 2 値ビットマップ画像の圧縮方法と

しては、MH (Modified Huffman)、MR (Modified  
READ)、MMR (Modified Modified READ)、JB  
IG (Joint Bi-level Image Group) 等があり、フ  
ァイルへの格納や FAX などに広く使用されている。

【0005】 これらの圧縮方法は、符号化処理が複雑で  
あることから、処理速度の高速化をねらった「ビットマ  
ップデータの圧縮方法及び圧縮装置」(特開平 8-51  
545 号) が発案されている。

【0006】 上記特開平 8-51545 号の画像圧縮方  
法では、まず画像を所定のライン数単位に水平方向に走  
査して行間のように連続する空白ラインを垂直スキップ  
として符号化し、空白ライン以外の領域を論理行として  
分離する。次に、分離した論理行を 1 列単位に垂直方向  
に走査して文字間のように黒画素を含まない空白領域を  
スキップし、空白領域以外の黒画素を含む領域を分離す  
る。

【0007】 そして、白画素のみからなる領域と、黒画  
素のみからなる領域をそれぞれ列数を指定してスキップ  
するように符号化し、同じ画素値の列が連続する領域を  
その値と繰り返し回数により符号化する。それ以外の領  
域は画素データをその列数と共にそのまま符号データと  
して出力する。

## 【0008】

【発明が解決しようとする課題】 前述のビットマップデ  
ータの圧縮方法には、ディザパターンのような一定の大  
きさのパターンが繰り返し出現する画像の圧縮率が低下  
するという問題があった。

【0009】 図 9 は、従来のディザパターンの符号化方  
法を説明する図である。同図上に示されるように、画像  
の走査は、例えば、8 ライン単位で行われる。空白行  
は、何も画像データがない部分であるので、8 ライン単  
位の行を何個スキップするかという情報を符号化する。

【0010】 ディザが含まれている行は論理行として認  
識され、8 ライン毎に、符号化が行われる。同図下は、  
ディザパターンの符号化方法の説明である。すなわち、  
従来では、ディザパターンはそのままビットパターンが  
符号化されており、例えば、8 ライン×N 画素からなる  
論理行の符号化は、ディザが含まれている部分領域をそ  
のまま符号化していた。従って、符号データは、符号  
語、画素幅 N、部分領域の画素データ (N バイト) から  
なり、N バイトよりも大きくなっていた。

【0011】 従って、ディザパターンを含むような画像  
であっても、有効に圧縮できる画像圧縮方法が望まれる  
ところである。本発明の課題は、連続して繰り返し出現  
するパターンを符号化することで圧縮率を向上させる符  
号化方法を提供することである。

## 【0012】

【課題を解決するための手段】 本発明の画像圧縮方法  
は、画像データを圧縮する方法であって、2 値画像デー  
タの内、反復パターンが含まれている行を抽出する抽出

10

20

30

40

50

ステップと、該反復パターンの単位パターンと、該単位パターンの繰り返し回数を検出する反復パターン検出ステップと、少なくとも、該単位パターンのデータと、反復パターン内における該単位パターンの繰り返し回数とを使って、画像データ内の反復パターンを含む行を符号化する符号化ステップとを備えることを特徴とする。

【0013】本発明によれば、ディザパターンのように、単位となるパターンが繰り返し出てくる画像データを有効に符号化し、画像データの圧縮率を向上することができる。

【0014】

【発明の実施の形態】図1は、本発明の実施形態に従った符号化方法の処理全体の流れを示すフローチャートである。

【0015】まず、入力された画像を所定ライン数単位に水平方向に走査し（ステップS101）、行間のように連続する空白ラインを垂直スキップとして符号化し、空白ライン以外の領域を論理行として分離する（ステップS102）。

【0016】次に、分離した論理行を1列単位に垂直方向に走査し（ステップS103）、文字間のように黒画素を含まない空白領域をスキップし、空白領域以外の黒画素を含む領域を分離する（ステップS104）。

【0017】更に、ステップS104の領域分離結果に基づいて、連続して繰り返し出現する同一幅及び同一画素値を有する反復パターンを分離する（ステップS105）。

【0018】そして、ステップS102、S104、S105で分離した各領域を要素として符号化を行う（S106）。図2は、本発明の実施形態の第一の原理を示す図である。

【0019】論理行の中から連続して繰り返し出現するパターンを検出し、繰り返しの基本となる反復パターンの長さとし、及びその繰り返し回数を使用して符号化することにより符号データサイズを削減する。これにより、符号データは、符号語、反復回数K、画素幅M、反復パターンの基本パターン（Mバイト）からなり、ディザパターンの符号データの長さが、ほぼ $N/K$ バイトとなり、従来のディザパターンの符号データの長さに比べ、略 $1/K$ の長さとなる。

【0020】画像の生成過程において、該画像の生成時に使用されるディザパターンの繰り返しの基本要素となるパターンのサイズが既知の場合がある。この場合、繰り返しパターンの検出をこの既知のサイズに限定して行うことで、全てのサイズについて検出を行う場合に比べて検出処理を大幅に削減し、高速に検出することができる。

【0021】図3は、本発明の実施形態に従った第二の原理を示す図である。検出した反復パターンをそのまま符号データとして出力する代わりに、反復パターン自体

をラインレンクス符号化等のデータ圧縮方法により圧縮することで、符号データサイズを更に削減することができる。

【0022】このとき、反復パターンの圧縮効果が無ければ、該反復パターンを圧縮しないで符号データとして出力する。同図においては、（A）に示されるようなディザパターンがあったとすると、反復長がM画素の反復パターンの基本パターンがK回繰り返されている。そこで、（B）に示されるように、M画素からなる反復パターンの基本パターンを取り出し、これをランレンクス符号化する。ランレンクス符号化の方法は、（C）に示されるように、基本パターンの先頭の8行画素からなる列から順に列の中のどこに黒画素があるかを調べる。まず、（B）の最初の4列は同じパターンであるので、1列分の画素パターンが4つ並んでいるという情報を生成する。次に、黒画素がない列が2列並んでおり、以下同様にして、各列毎に、基本となる黒画素のパターンとそのパターンが列方向に何回連続するかの情報を生成し、M' バイトのランレンクス符号データを生成する。ランレンクス符号化を行うと、行わないよりも更に良い圧縮効果が得られる。

【0023】ランレンクス符号化しない場合は、符号データは、符号語、ランレンクスフラグ=0（ランレンクス符号化していないことを示す）、パターン長M、反復回数K、パターンデータからなる。ランレンクス符号化すると、符号語、ランレンクスフラグ=1（ランレンクス符号化していることを示す）、ランレンクス符号データ長M' バイト、反復回数K、ランレンクス符号データという構成の符号データが得られる。

【0024】図4は、本発明の実施形態の処理の流れを示すフローチャートである。同図のフローチャートは、図1に示した原理図におけるステップS105とステップS106の処理に対応し、分離された論理行を符号化する処理を示している。空白領域の符号化は従来通りの方法で行う。

【0025】以下に、処理フローの説明をする。

ステップS201：符号化処理開始前に必要な、作業用メモリ領域の割当てや、内部パラメータの初期化などの初期化処理を行う。後述するパターンテーブルに登録されているパターンの個数をカウントするパラメータもここでゼロに初期化しておく。

【0026】ステップS202：論理行から連続して繰り返し出現する同一サイズ及び画素値を持つパターンの有無を検出する。

ステップS203：S202の検出結果に基づき、反復パターンが検出された場合には処理をS205に進み、反復パターンが検出されなかった場合には処理をS204に進む。

【0027】ステップS204：従来の符号化を行う。

ステップS205：検出された反復パターンの領域を符

10

20

30

40

50

号化する。

ステップS206: 全ての論理行の符号化が完了したか否かを判断し、完了していない場合には処理をS202に戻し、完了した場合には符号化処理を終了する。

【0028】以下、反復パターン検出処理S202と反復パターン符号化S205の各処理について、図面を参照しながら詳細に説明する。図5は、反復パターン検出の実施形態を示す処理フローであり、図4のS202の処理に対応する。

【0029】論理行内を1列単位に走査して、連続する反復パターンの単位パターンの有無を確認し、反復パターンの単位パターンがある場合には反復パターンの単位パターンと共にその長さlenと反復回数numを取得する。

【0030】ステップS301: 検出された反復パターンの単位パターンの反復回数を示すnumに0を設定する。また、反復パターンの単位パターンの検出を開始する列を示す変数aに1、すなわち論理行の先頭列を設定する。

【0031】ステップS302: 検出を開始する反復パターンの長さとしてlenにMINLENを設定する。MINLENは、予想される反復パターンの単位パターンから適切な値を割り当てればよい。例えば、画像中の反復パターンの単位パターンの幅が4画素以上であることが予め分かっている場合には、MINLEN=4とする。こうすることで、存在しない反復パターンの検出処理を回避し、処理時間を短縮できる。

【0032】ステップS303: 参照する列番号を示す変数bに、検出する反復パターンの単位パターン長さlenをaに加えた値を設定する。

ステップS304: aを先頭とするlen列の画像と、bを先頭とするlen列の画像を比較し、一致するか否かを判断する。両者が一致しない場合には処理をステップS305に移し、両者が一致する場合には処理をステップS310に進める。

【0033】ステップS305: 長さlen列の検出する反復パターンの単位パターンの長さlenに1を加えて、検出する反復パターンの単位パターンの長さを1列分長く設定する。

【0034】ステップS306: 反復パターンの単位パターンの長さlenが検出する最大の長さMAXLENに達したか、またはlenが検出処理を適用していない残り列数の半分に達したか否かを確認し、いずれかの条件を満たす場合には処理をステップS303に戻し、両条件ともに満たさない場合には処理をステップS307に進める。ここで、lenが検出処理を適用していない残り列数(WIDTH-a)の半分に達するということは、検出処理しようとしている列位置aから始まる長さlenの単位パターンが、残りの列の中に1つより多くは含まれ得ないことを意味している。従って、反復パ

ターンとはならないので、次のパターン検出処理であるステップS307に進むのである。

【0035】ステップS307: 反復パターン検出開始列を示すaに1を加える。すなわち、反復パターンの単位パターンの検出を開始する列を1列進める。

ステップS308: 検出処理を適用していない残り列数が、検出する反復パターンの単位パターンの最短長の2倍以上残っている場合には処理をステップS302に戻し、そうでない場合にはステップS309に処理を進める。ステップS309に進む場合は、検出しようとする単位パターンが残りの列数の中に1つより多く含まれないことになるので、反復パターンとはならないことから、反復パターンの検出処理を終了するための処理に移るものである。

【0036】ステップS309: 論理行内に連続して出現する反復パターンが存在しない旨出力して処理を終了する。以上の処理をステップS304でYESとなるまで繰り返し行うことにより、論理行内の連続して出現する反復パターンの有無と、反復パターンが存在した場合には、その位置及び反復パターンの単位パターンが検出された場合には更に以下の処理を行う。

【0037】ステップS310: ここまでの処理により反復パターンが2個連続していることが検出されていることになるため、反復パターンの反復回数をカウントする変数numに2を設定する。

【0038】ステップS311: 反復パターン有無を検出する次の列を示すようにbをlenに加算し、1パターン分の列数だけ先に進める。

ステップS315: 次に、1パターン先の検出処理開始位置である列bが、列幅WIDTHからパターンの幅であるlenを引いたものより小さいか否か、すなわち、列bから始まる長さlenのパターンが残りの列数の中に収まるか否かを判断する。bがWIDTH-len以下である場合には、ステップS312に進み、bがWIDTH-lenより大きい場合には、ステップS314に進む。

【0039】ステップS312: S304と同様に、aを先頭とするlen列の画像と、bを先頭とするlen列の画像を比較し、一致するか否かを判断する。両者が一致しない場合には処理をS314に移し、両者が一致する場合には処理をS313に進める。

【0040】ステップS313: 検出した反復パターンの反復回数を示すカウンタnumに1を加算し、処理をS311に戻す。連続する反復パターンが検出されなくなるまでステップS311～S313を繰り返し行うことで、反復パターンの連続する個数が検出される。

【0041】ステップS314: 検出された反復パターンの開始位置aと、反復パターンの長さlen及び反復回数numを結果として出力し、処理を終了する。以上の処理により、論理行内に反復パターンが存在するか否

かが判明し、反復パターンが存在する場合には、その開始位置と反復パターンの単位パターンの長さ及び反復回数が得られる。

【0042】図6は、符号語のビット構成の実施形態を示す図である。図6(a)は、検出した反復パターンをランレングス符号化しないで反復パターンをそのまま符号として出力する例、図6(b)は反復パターンをランレングス符号化して出力する場合の例である。図6

(a)では、ランレングス符号データでないことを示すため、ランレングス符号化フラグを0にセットし、単位パターンの長さを示すデータ長Mで指定された画素幅の反復パターンの単位パターンデータを出力する。図6

(b)では、ランレングス符号化されていることを示すため、ランレングス符号化フラグを1にセットし、ランレングス符号化したデータの長さを示すデータ長M'バイトのランレングス符号データを出力する。何れの場合も、符号化データの中には、反復パターンの単位パターンが何回繰り返し出てくるかを示す反復回数Kが設定される。

【0043】図7は、パターンテーブルの例を示した図である。前述のパターンテーブルは、ディザパターンを符号化する場合に、例えば、1ページの文書の中で出てきた単位パターンをその長さと共に登録し、1ページの中で同じ単位パターンが現れた時には、前述の符号化データの内のパターンデータあるいは、ランレングス符号データの部分に、パターンテーブルの登録番号を格納するようにする。登録番号は、画像を符号化していく間に、先に出てきた単位パターンから順次設定されるものであり、最初に出てきた単位パターンが登録番号1であり、2番目に出てきた単位パターンが登録番号2という具合である。

【0044】このパターンテーブルは、画像を符号化する場合に生成されるが、符号化データの中には含まれない。符号化された画像を復号する場合には、符号データから順次単位パターンを読み取り、パターンテーブルを復号側で再生する様にする。すると、符号データ内には、符号化の時に出てきた順番で単位パターンが埋め込まれているので、復号する場合にも、順次符号データを復号しながらパターンテーブルを作成すると、符号化の際に得られたものと同じものが得られることになる。したがって、符号データの中に、パターンテーブルの登録番号が出てきた場合には、復号の際に生成したパターンテーブルの登録番号を参照して、単位パターンを取得すれば、正しく符号データを復号することができる。

【0045】以上のように、パターンテーブルを使用することにより、符号データに単位パターンのビットデータをそのまま入れるより、圧縮率を上げることができる。更に、ランレングス符号化と併用すれば、画像データの圧縮率を更に上げることができる。

【0046】図8は、本実施形態の方法をプログラムで

実現する場合に情報処理装置に必要とされるハードウェア環境図である。バス10でCPU11と接続されたROM12には、BIOSなどの基本プログラムが格納される。また、当該プログラムは、バス10でCPU11と接続された、ハードディスクなどの記憶装置17に格納され、RAM13にコピーされて、CPU11によって実行される。読み取り装置18は、フレキシブルディスク、CD-ROM、DVDなどの可搬記録媒体19から当該プログラムを読み取り、情報処理装置21にインストールを可能とするものである。あるいは、可搬記録媒体19から当該プログラムを直接読み込んで、CPU11に実行させても良い。

【0047】入出力装置20は、キーボード、マウス、テンプレート、ディスプレイなどからなり、情報処理装置21のユーザからの命令をCPU11に伝えたり、CPU11の演算結果をユーザに提示するためのものである。

【0048】通信インターフェース14は、情報処理装置21をネットワーク15を介して情報提供者16に接続するものである。情報提供者16は、当該プログラムを持っていて、これをネットワーク15経由で情報処理装置21にダウンロードし、インストールして、CPU11が実行しても良い。あるいは、情報提供者16と情報処理装置21とをネットワーク15で接続したまま、ネットワーク環境のもとで、当該プログラムを実行するようにしても良い。

【0049】(付記1)画像データを圧縮する方法であって、2値画像データの内、反復パターンが含まれている行を抽出する抽出ステップと、該反復パターンの単位パターンと、該単位パターンの繰り返し回数を検出する反復パターン検出ステップと、少なくとも、該単位パターンのデータと、反復パターン内における該単位パターンの繰り返し回数を使って、画像データ内の反復パターンを含む行を符号化する符号化ステップと、を備えることを特徴とする画像圧縮方法。

【0050】(付記2)前記単位パターンの画素幅が予め定められた1以上の固定値であることを特徴とする付記1に記載の画像圧縮方法。

(付記3)前記符号化ステップにおいては、単位パターンを更にデータ圧縮し、該圧縮された単位パターンと該単位パターンの繰り返し回数により符号化することの特徴とする付記1に記載の画像圧縮方法。

【0051】(付記4)前記符号化ステップは、ランレングス符号化方法を使用することを特徴とする付記1に記載の画像圧縮方法。

(付記5)前記符号化ステップにおいて、出現した単位パターンを登録するテーブルを生成し、該単位パターンと同じ単位パターンが符号化の過程で2回以上出てくる場合には、該単位パターンの代わりに該テーブルにおける該単位パターンの登録番号を用いて、画像データを符

号化することを特徴とする付記1に記載の画像圧縮方法。

【0052】（付記6）画像データを圧縮する方法であって、2値画像データの内、反復パターンが含まれている行を抽出する抽出ステップと、該反復パターンの単位パターンと、該単位パターンの繰り返し回数を検出する反復パターン検出ステップと、少なくとも、該単位パターンのデータと、反復パターン内における該単位パターンの繰り返し回数とを使って、画像データ内の反復パターンを含む行を符号化する符号化ステップと、を備えることを特徴とする画像圧縮方法を情報処理装置に実現させるプログラム。

【0053】（付記7）画像データを圧縮する方法であって、2値画像データの内、反復パターンが含まれている行を抽出する抽出ステップと、該反復パターンの単位パターンと、該単位パターンの繰り返し回数を検出する反復パターン検出ステップと、少なくとも、該単位パターンのデータと、反復パターン内における該単位パターンの繰り返し回数とを使って、画像データ内の反復パターンを含む行を符号化する符号化ステップと、を備えることを特徴とする画像圧縮方法を情報処理装置に実現させるプログラムを格納した、情報処理装置読み取り可能な記録媒体。

【0054】（付記8）画像データを圧縮する装置であって、2値画像データの内、反復パターンが含まれている行を抽出する抽出手段と、該反復パターンの単位パターンと、該単位パターンの繰り返し回数を検出する反復パターン検出手段と、少なくとも、該単位パターンのデータと、反復パターン内における該単位パターンの繰り返し回数とを使って、画像データ内の反復パターンを含む行を符号化する符号化手段と、を備えることを特徴とする画像圧縮装置。

【0055】

【発明の効果】以上説明したように、本発明の画像圧縮方法によれば、ディザパターンのように同じパターンが連続して繰り返し出現する画像から繰り返しの基本とな

る基本パターンを検出して符号化することにより、写真などのディザを含む画像の圧縮率が向上する。このため、データを格納するための記憶容量の削減やデータ転送時間の短縮に大きく寄与する。

【図面の簡単な説明】

【図1】本発明の実施形態に従った符号化方法の処理全体の流れを示すフローチャートである。

【図2】本発明の実施形態の第一の原理を示す図である。

【図3】本発明の実施形態に従った第二の原理を示す図である。

【図4】本発明の実施形態の処理の流れを示すフローチャートである。

【図5】反復パターン検出の実施形態を示す処理フローである。

【図6】符号語のビット構成の実施形態を示す図である。

【図7】パターンテーブルの例を示した図である。

【図8】本実施形態の方法をプログラムで実現する場合に情報処理装置に必要とされるハードウェア環境図である。

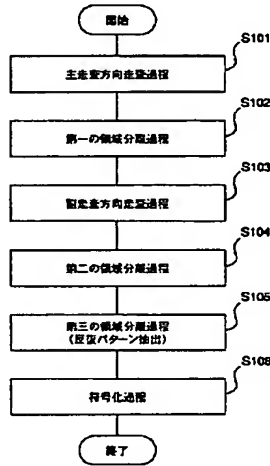
【図9】従来のディザパターンの符号化方法を説明する図である。

【符号の説明】

10	バス
11	CPU
12	ROM
13	RAM
14	通信インターフェース
15	ネットワーク
16	情報提供者
17	記憶装置
18	読み取り装置
19	可搬記録媒体
20	入出力装置
21	情報処理装置

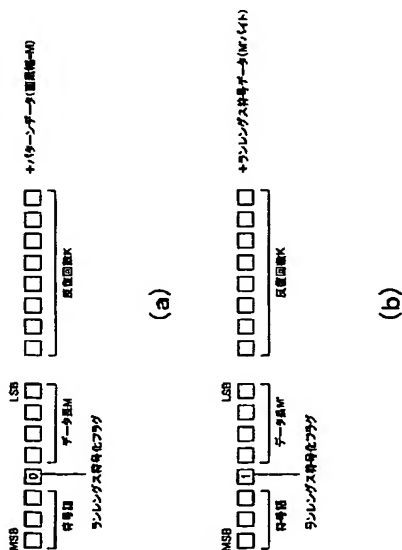
【図1】

本発明の実施形態に従った符号化方法の処理全体の流れを示すフローチャート



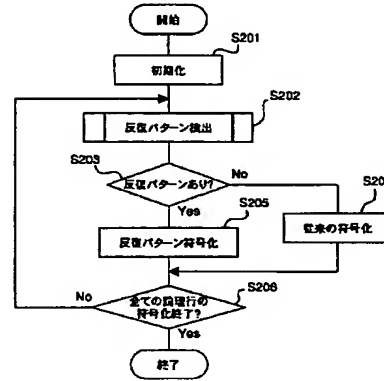
【図6】

符号語のビット構成の実施形態を示す図



【図4】

本発明の実施形態の処理の流れを示すフローチャート



【図7】

パターンテーブルの例を示した図

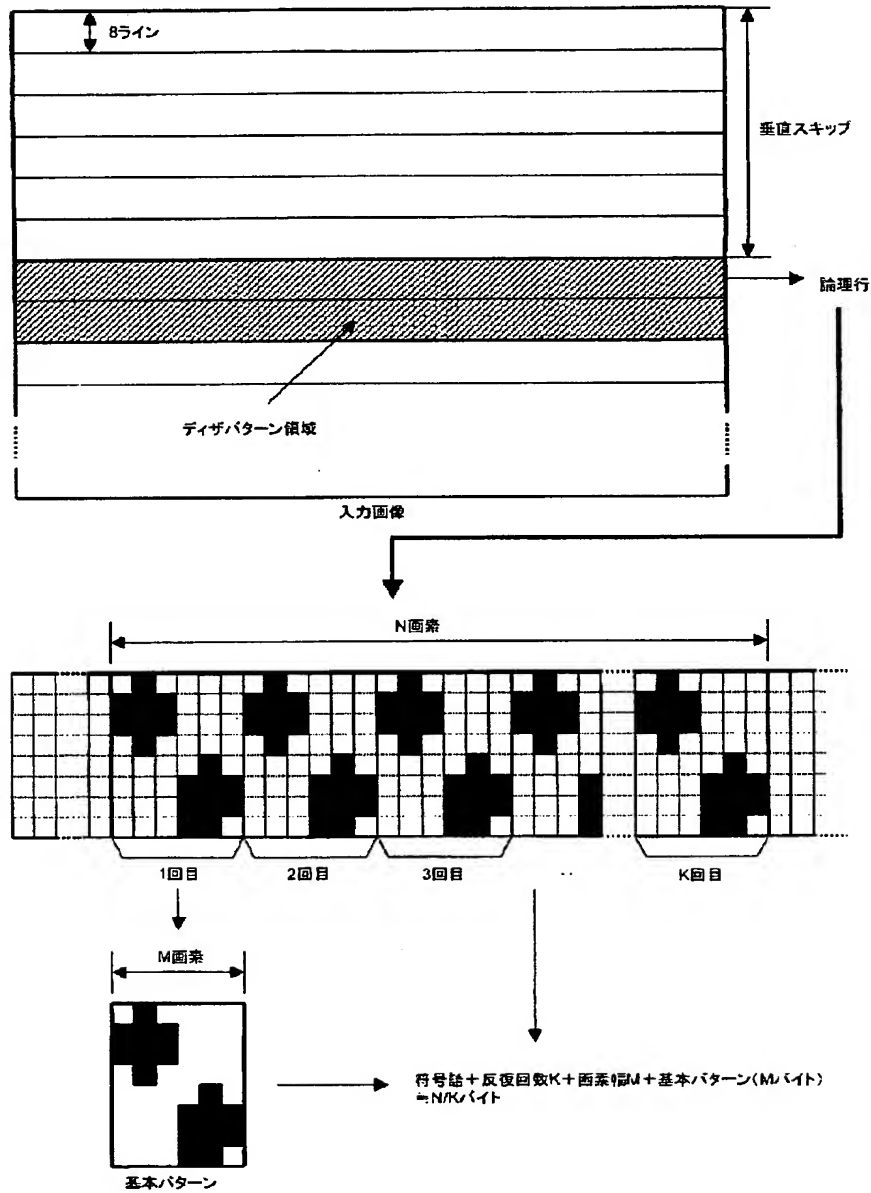
単位パターンの長さ(図2, 図3ではM画素)

No	長さ	反復パターンの単位パターン
1	w1	単位パターン1
2	w2	単位パターン2
...		



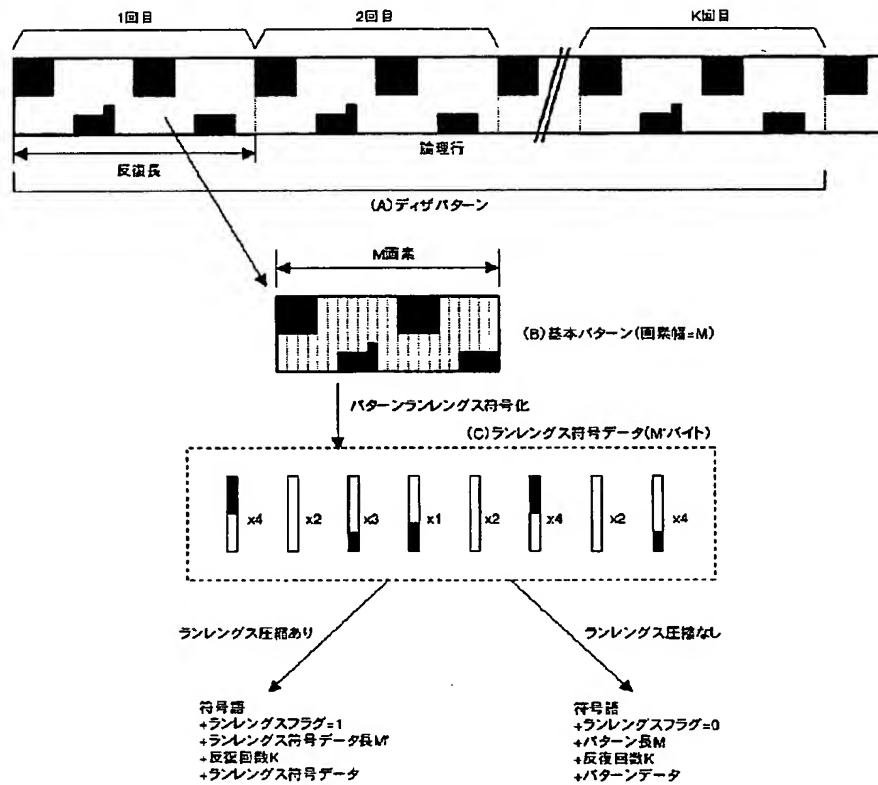
【図2】

## 本発明の実施形態の第一の原理を示す図



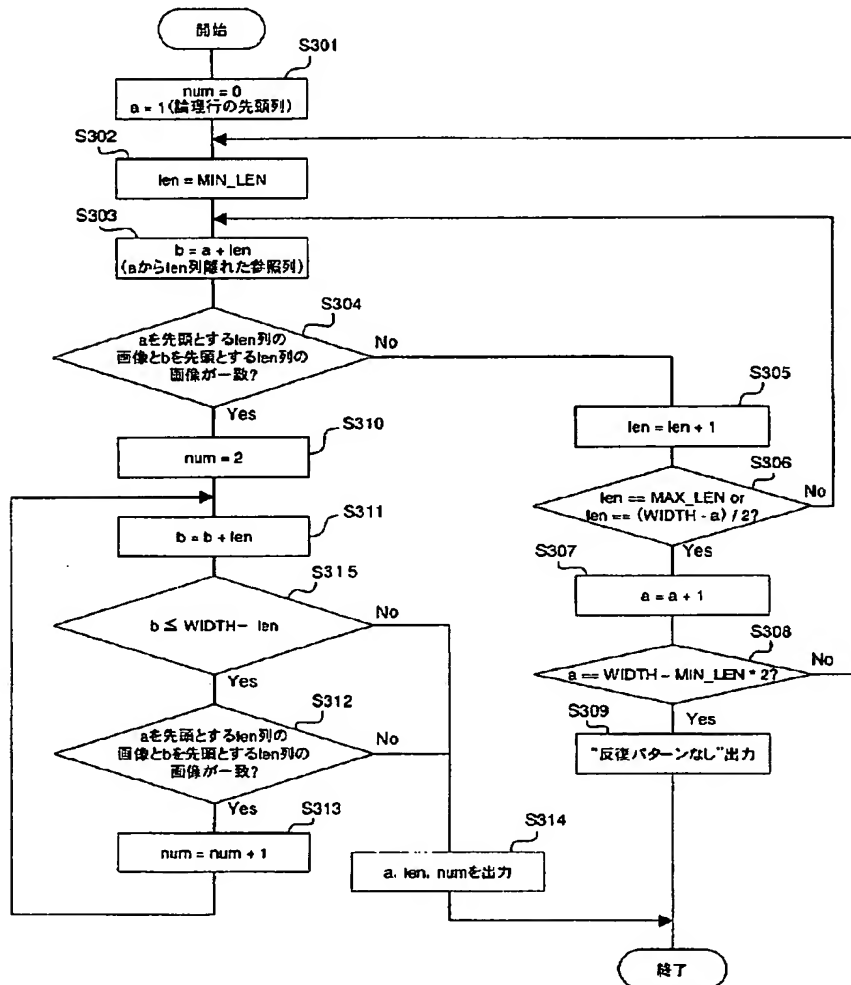
【図3】

本発明の実施形態に従った第二の原理を示す図



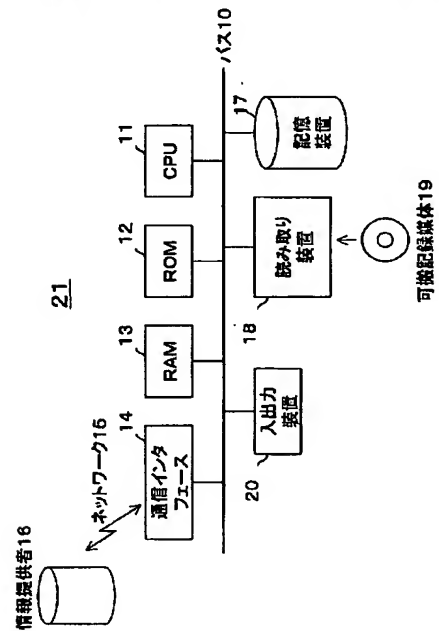
【図5】

## 反復パターン検出の実施形態を示す処理フロー



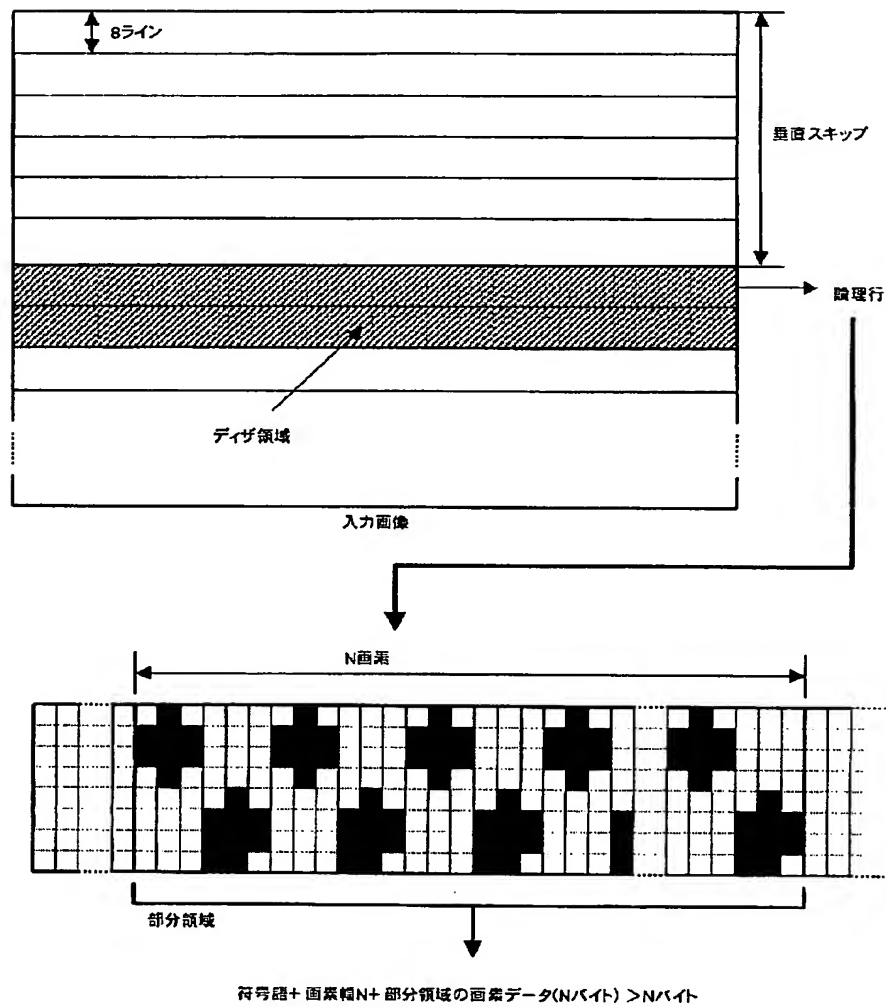
【図8】

本実施形態の方法をプログラムで実現する場合に  
情報処理装置に必要とされるハードウェア環境図



【図9】

## 従来のディザパターンの符号化方法を説明する図



フロントページの続き

Fターム(参考) 5C078 AA01 AA06 BA35 CA01 DA01  
DA05  
5J064 AA01 BA08 BB05 BC01 BC05  
BC14 BC29 BD02